Operating System Concepts

Summary

Al-Iraqia University
Network Engineering

Chapter 1

What's Operating System ?

Operating System : is a program which acts as interface between user of a computer and the computer hardware, it is important part of almost every computer system .

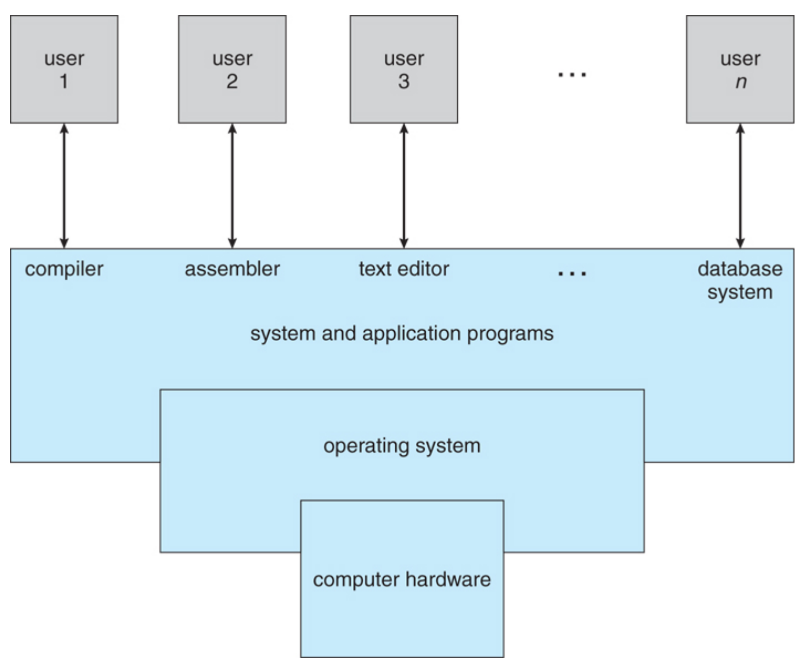
 # The purpose of OS : is to provide an environment in which a user make execute programs .

 # The goals of OS :

1- Primary goal : Make the computer system convenient to use .
2- Secondary goal : Use the computer hardware in efficient mannar .

 # The component of OS :

1- Hardware(CPU).
2- Operating System .
3- Applications.
4- Users.



Q : What operating system do ?

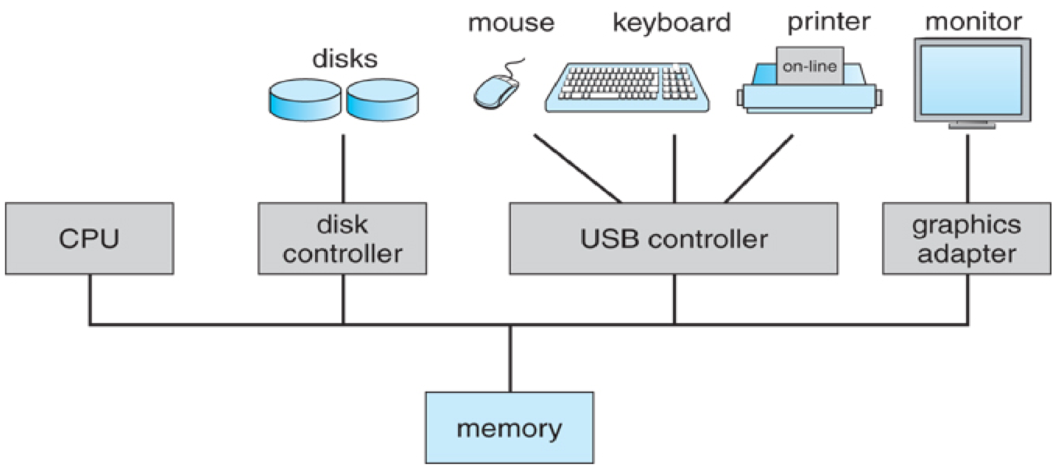
A:

1- Depends on the point of view.
2- Users want convenience, ease of use don’t care about resource utilization.
3- But shared computer such as mainframe or minicomputer must keep all users happy.
4- Handheld computers are resource poor, optimized for usability and battery life.
5- Some computers have little or on user interface, such as embedded computers in devices and automobiles.

Computer System Organization :

A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access to shared memory, Each device controller is in charge of a specific type of device (for example, disk drives, audio devices, or video displays). The CPU and the device controllers can execute in parallel, competing for memory cycles. To ensure orderly access to the shared memory,a memory controller synchronizes access to the memory.
For a computer to start running—for instance, when it is powered up or rebooted—it needs to have an initial program to run. This initial program, or bootstrap program, tends to be simple. Typically, it is stored within the computer hardware in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term firmware. It initializes all aspects of the system, from CPU registers to device controllers to memory contents. The bootstrap program must know

how to load the operating system and how to start executing that system. To accomplish this goal, the bootstrap program must locate the operating-system kernel and load it into memory.

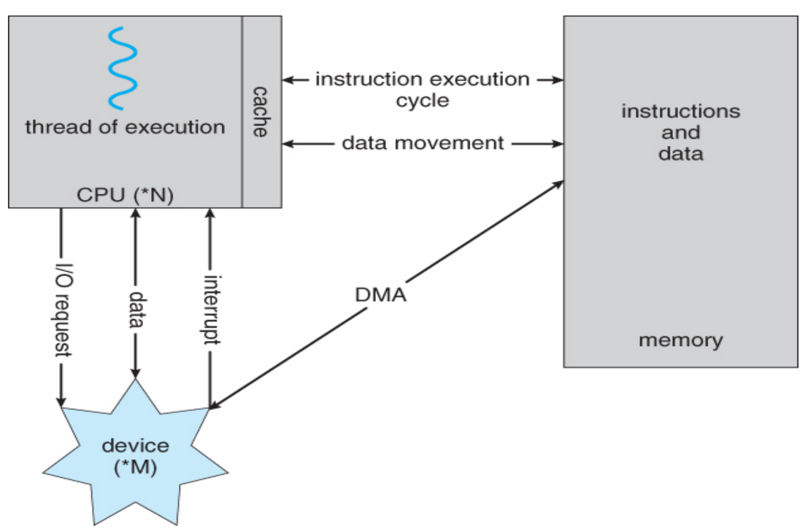


- ➢ <u>Storage Structure</u>
- ➢ <u>I\O Structure</u>

<u>Computer System Architecture :</u>

A computer system can be organized in a number of different ways, which we can categorize roughly according to the number of general-purpose processors used.

Figure : How a modern computer system works.



➢ Single-Processor Systems :

Until recently, most computer systems used a single processor. On a single processor system, there is one main CPU capable of executing a general-purpose instruction set, including instructions from user processes. Almost all single processor systems have other special-purpose processors as well. They may come in the form of device-specific processors, such as disk, keyboard, and graphics controllers; or, on mainframes, they may come in the form of more general-purpose processors, such as I/O processors that move data rapidly among the components of the system.

➢ Multiprocessor Systems :

Within the past several years, multiprocessor systems (also known as parallel systems or multicore systems) have begun to dominate the landscape of computing. Such systems have two or more processors in close communication,sharing the computer bus and sometimes the clock, memory, and peripheral devices. Multiprocessor systems first appeared prominently appeared in servers and have since migrated to desktop and laptop systems. Recently,multiple processors have appeared on mobile devices such as smartphones and tablet computers.

Multiprocessor systems have three main advantages:

1. **Increased throughput:** By increasing the number of processors, we expect to get more work done in less time.
2. **Economy of scale:** Multiprocessor systems can cost less than equivalent multiple single-processor systems, because they can share peripherals, mass storage, and power supplies.

3. **Increased reliability:** If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down.

Operating System Operation :

As mentioned earlier, modern operating systems are interrupt driven. If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen. Events are almost always signaled by the occurrence of an interrupt or a trap. A trap (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed. The interrupt-driven nature of an operating system defines that system’s general structure. For each type of interrupt, separate segments of code in the operating system determine what action should be taken. An interrupt service routine is provided to deal with the interrupt. Since the operating system and the users share the hardware and software resources of the computer system, we need to make sure that an error in a user program could cause problems only for the one program running. With sharing, many processes could be adversely affected by a bug in one program. For example, if a process gets stuck in an infinite loop, this loop could prevent the correct operation of many other processes. More subtle errors can occur in a multiprogramming system, where one erroneous program might modify another program, the data of another program, or even the operating system itself.

- ### Dual-Mode and Multimode Operation:
- ### Timer :

Process Management :

❖ OBJECTIVES :

• To introduce the notion of a process—a program in execution, which forms
the basis of all computation.
• To describe the various features of processes, including scheduling, creation, and termination.
• To explore interprocess communication using shared memory and message passing.
• To describe communication in client – server systems.

➢ Process Concept :

A question that arises in discussing operating systems involves what to call all the CPU activities. A batch system executes jobs, whereas a time-shared system has user programs, or tasks. Even on a single-user system, a user may be able to run several programs at one time: a word processor, a Web browser, and an e-mail package. And even if a user can execute only one program at a time, such as on an embedded device that does not support multitasking, the operating system may need to support its own internal programmed activities, such as memory management. In many respects, all these activities are similar, so we call all of them processes.

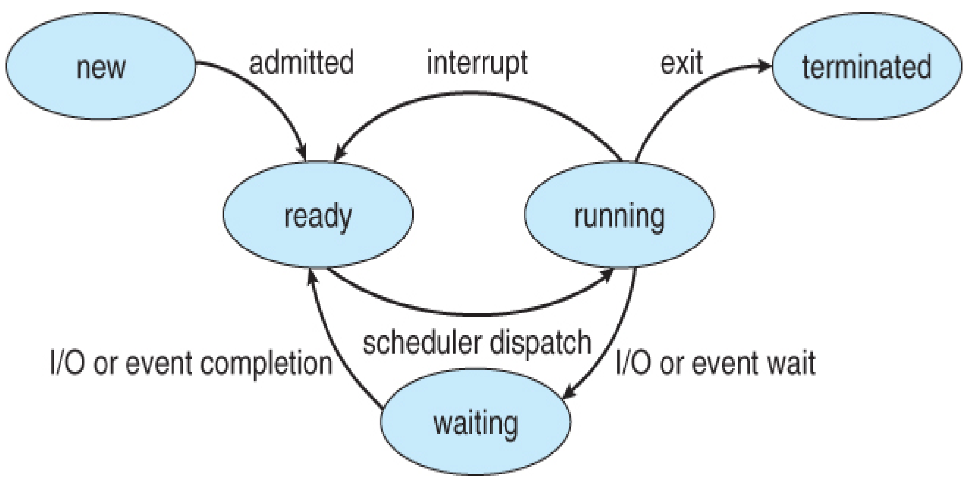
The terms job and process are used almost interchangeably in this text. Although we personally prefer the term process, much of operating-system theory and terminology was developed during a time when the major activity of operating systems was job processing. It would be misleading to avoid the use of commonly accepted terms that include the word job (such as job scheduling) simply because process has superseded job.

➢ Process State :

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. A process may be in one of the following states:

- **New:** The process is being created.
- **Running:** Instructions are being executed.
- **Waiting:** The process is waiting for some event to occur (such as an I/O completion or reception of a signal).
- **Ready:** The process is waiting to be assigned to a processor.
- **Terminated:** The process has finished execution.

These names are arbitrary, and they vary across operating systems. The states that they represent are found on all systems, however. Certain operating systems also more finely delineate process states. It is important to realize that only one process can be running on any processor at any instant. Many processes may be ready and waiting, however. The state diagram corresponding to these states is presented in Figure :



Memory Management :

memory is central to the operation of a modern computer system. Memory consists of a large array of bytes, each with its own address. The CPU fetches instructions from memory according to the value of the program counter. These instructions may cause additional loading from and storing to specific memory addresses.
A typical instruction-execution cycle, for example, first fetches an instruction from memory. The instruction is then decoded and may cause operands to be fetched from memory. After the instruction has been executed on the operands, results may be stored back in memory. The memory unit sees only a stream of memory addresses; it does not know how they are generated (by the instruction counter, indexing, indirection, literal addresses, and so on) or what they are for (instructions or data). Accordingly, we can ignore how a program generates a memory address. We are interested only in the sequence of memory addresses generated by the running program.

The operating system is responsible for the following activities in Connection with memory management:

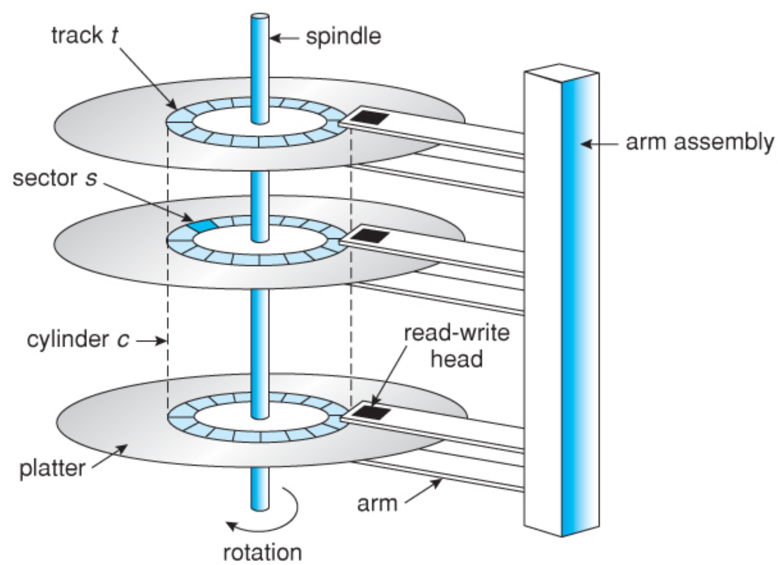
- Keeping track of which parts of memory are currently being used and who is using them.
- Deciding which processes (or parts of processes) and data to move into and out of memory.
- Allocating and deallocating memory space as needed.

Storage Management :

To make the computer system convenient for users, the operating system provides a uniform, logical view of information storage. The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file. The operating system maps files onto physical media and accesses these files via the storage devices.

- **Magnetic Disks:**

  Magnetic disks provide the bulk of secondary storage for modern computer systems. Conceptually, disks are relatively simple fig.
  Each disk platter has a flat circular shape, like a CD. Common platter diameters range from 1.8 to 3.5 inches. The two surfaces of a platter are covered with a magnetic material. We store information by recording it magnetically on the platters.



- **Disk Structure**
- **Disk Attachment**
- **Disk Management**
- **Swap-Space Management**